An Analysis of Edge Detectors and Improved Hybrid Technique for Edge Detection of Images

Mamta. Juneja, and Parvinder S. Sandhu

***Abstract*—**This paper provides an analysis of various edge detection techniques viz. sobel, prewitt, robert, laplacian of gaussian and canny in which canny edge detector outperforms all existing edge detectors. A new hybrid edge detection technique for extracting edges, line, boundaries and circles from a given input image is also proposed in this paper. It utilizes canny for extracting edges and thereafter output edges are refined by application of hough transform. It has been successful to extract edges, lines, boundaries and circles of any image and is immune to various noise effects and other disturbances.

***Keywords*—**canny, circle detection**,** edge detection, hough transform, laplacian of gaussian, line detection, prewitt, robert, sobel.

I. Introduction

An edge [1] is defined to be a sequence of pixels that border two homogeneous regions of different intensities which help in segmentation and object recognition. It indicates discontinuity in image intensity function. An Edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either Step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance. However, Step and Line edges are rare in real images. Because of low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become Ramp Edges and Line Edges become Roof edges, where intensity changes are not instantaneous but occur over a finite distance. Illustrations of these edge shapes are shown in Figure 1.

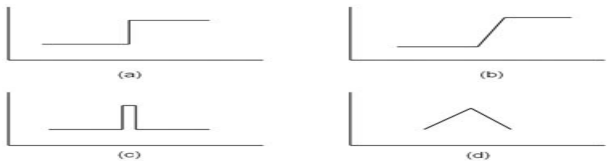

Fig. 1 Types of Edges a) Step b) Ramp c) Line d) Roof

Mamta.Juneja is Assistant Professor in University Institute of Engineering and technology, Panjab Univesity, Chandigarh. E-mail: er_mamta@yahoo. com.

Prof. Dr. Parvinder S. Sandhu is with Rayat and Bahra Institute of Engineering and Bio-Technology, Punjab, India.

A. Edge Detection [2]

It is technique for locating areas with strong intensity contrasts. It amounts to locating each edge pixel and determining its orientation. Edge detection is one of the most commonly used operations in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. Since computer vision involves the identification and classification of objects in an image, edge detections is an essential tool. Edge detection is a very important area in the field of Computer Vision, Segmentation and object recognition. It is a fundamental of low-level image processing and good edges are necessary for higher level processing.

Edge detectors are the filters which filter out the useless information and preserve the useful information in image. Three fundamental steps involved in edge detection are:

1. Image smoothing for noise reduction: this step involve filtering the image for improving the performance of edge detector.

2. Detection: This step involves extracting all edge points that are potential candidates to become edge point.

3. Edge localization: This step involves selecting from the candidate edge points only the points that are true members of set of points comprising an edge.

Edge detection techniques are broadly divided in to two categories gradient edge detection and Laplacian edge detection.

Gradient based Edge Detection: This method detects the edge points in the image by using the gradient. The gradient is tool for finding the edge strength and direction at location, f, and is denoted by Δf , and defined as the vector,

$$\Delta f = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

This vector always points in the direction of greatest rate of change at location (x, y). The magnitude of the vector can be given as,

$$M(x,y) = \sqrt{\left(G_x^2 + G_y^2\right)} \quad (2)$$

The direction of the gradient vector is given by an angle, θ(x, *y*). Consider the one dimensional signal with an edge shown by jump in intensity as shown in Figure 2 below:

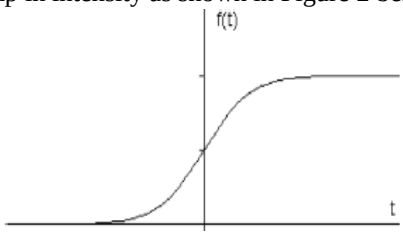

Fig. 2 A Signal Representation

If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to t) we get the shape shown in Figure 3:

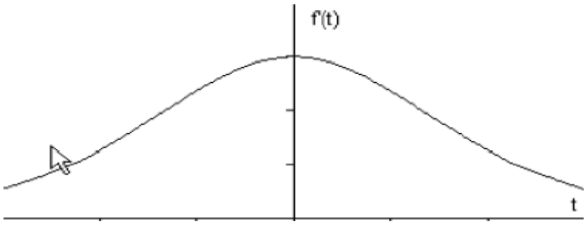

Fig. 3 First Derivative of Signal [2]

The first derivative of the one dimensional signal clearly shows maximum is located at the center of the edge in the original signal.

Laplacian based Edge Detection: This method basically searches for zero crossings in the second derivative of the image. If the first derivative is maximum, then its second derivative will definitely be zero. This phenomenon gives rise to alternate method for edge detection which is known as Laplacian. Hence, we can detect the edges by searching for zero crossings in the 2nd derivative, Figure 4 shows second derivative of the signal.

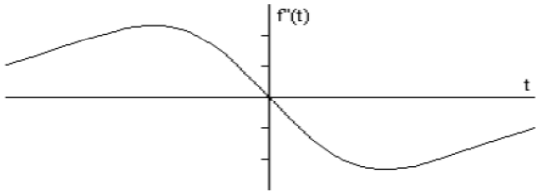

Fig. 4 Second Derivative of Signal [2]

B. Performance Criterion for Edge Detectors

There are problems of false edge detection, missing true edges, producing thin or thick lines and problems due to noise. Hence, it was necessary to define certain performance criterion for edge detectors. Performance criterion for edge detectors involve following points.

Good detection: good detection means the probability of falsely marking the non-edge points should be as low as possible.

Good localization: The edge detected should be as close as to the real edge points.

Single response to single point: The edge detector must be able to give single response to single edge point.

C. Edge detectors

The various popular edge detectors used in image processing as described in [1] are as follows:

a) Sobel Edge Detector: The Sobel edge detection method is introduced by Sobel in 1970 [1]. In Sobel edge detector, the task of edge detection is fulfilled by performing a 2D spatial gradient convolution operation on an image. This operator uses two convolution masks Gx and Gy.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3)$$

Here, Gx and Gy are computed as,

$$G_x = -1z_1 + 1z_3 - 2z_4 + 2z_6 - 1z_7 + 1z_9 \quad (4)$$

$$G_{yx} = 1z_1 + 2z_2 + 1z_3 - 1z_7 - 2z_8 - 1z_9 \quad (5)$$

Where Zi, i=1,2,.,9 are intensity levels of each pixel in the convolution are

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix} \quad (6)$$

Finally, the gradient magnitude is thresholded. Sobel operator is very simple and effective way for finding the edges in image. The sobel operator is used mostly for detecting horizontal and vertical edges. However the edges detected by this sobel operator are thick which may not be suitable for some applications where the detection of the outer most contour of an object is required.

b) Prewitt's Edge Detector: The Prewitt edge detection is proposed by Prewitt in 1970) [1] to estimate the magnitude and orientation of an edge. The Prewitt's edge detector also uses the three by three convolution mask, which is little different from that Sobel edge detector. The convolution masks smoothens the image. The convolution mask is:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (7)$$

The prewitt's edge detector is mainly used for detecting vertical and horizontal edges in image.

c) Robert's Cross Operator: The Roberts edge detection is introduced by Lawrence Roberts in 1965.The Roberts cross operator has 2x2 convolutions mask. The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. The convolution masks for Roberts operator is shown below.

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \quad (8)$$

The Roberts cross detector has only one disadvantage that it fails to detect few of the edges. These masks are mainly designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations.

d) Laplacian of Gaussian (LoG): The Laplacian of Gaussian was proposed by Marr in 1982.The Laplacian is a 2-

D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. Laplacian is applied to the image for smoothing purpose, in order to reduce the sensitivity to noise.

The Laplacian L(x, y) of an image with pixel intensity values I(x, y) is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (9)$$

and convolution mask are:

	-1	
-1	4	-1
	-1	

1	1	1
1	8	1
1	1	1

(10)

The LoG kernel can be pre calculated in advance so only one convolution needs to be performed at run-time on the image. The 2-D LoG function centered on zero and with Gaussian standard deviations has the form:

$$LoG(x,y) = \frac{-1}{\pi\sigma^4}\left[1 - \left(\frac{x^2+y^2}{2\sigma^2}\right)\right] e^{\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (11)$$

e) Canny edge detector: The Canny edge detector is widely considered to be the standard edge detection algorithm in the industry. It was first created by John Canny for his Master's thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. Canny saw the edge detection problem as a signal processing optimization problem, so he developed an objective function to be optimized as illustrated in [3-4].The solution to this problem was a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. The objective function was designed to achieve the following optimization constraints:

- Maximize the signal to noise ratio to give good detection: This favors the marking of true positives.
- Achieve good localization to accurately mark edges.
- Minimize the number of responses to a single edge: This favors the identification of true negatives, that is, non-edges are not marked.

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It has been used as it optimal edge detection algorithm. In this situation, an optimal edge detector means:
Good Detection: It should mark as many real edges in the image as possible.

Good Localization: Edges marked should be as close as possible to the edge in the real image.

Minimal Response: Edge in the image should only be marked once, and where possible, image noise should not create false edges.

E. Comparison of Edge Detectors [5]

Gradient-based edge detection algorithms such as the Prewitt filter have a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be adapted to a given image. An adaptive edge-detection algorithm is necessary to provide a robust solution that is adaptable to the varying noise levels of these images to help distinguish valid image contents from visual artifacts introduced by noise. The performance of the Canny algorithm depends heavily on the adjustable parameters, σ, which is the standard deviation for the Gaussian filter, and the threshold values, 'T1' and 'T2'. σ also controls the size of the Gaussian filter. The bigger the value for σ, the larger the size of the Gaussian filter becomes. This implies more blurring, necessary for noisy images, as well as detecting larger edges. As expected, however, the larger the scale of the Gaussian, the less accurate is the localization of the edge. Smaller values of σ imply a smaller Gaussian filter which limits the amount of blurring, maintaining finer edges in the image. The user can tailor the algorithm by adjusting these parameters to adapt to different environments.

Canny edge detection algorithm is computationally more expensive compared to Sobel, Prewitt and Robert's operator. However, the canny edge detection algorithm performs better than all these operators under almost all scenarios as per survey carried in [6].

F. Hough Transform (HT)

The Hough transform was introduced by Hough [7] with his patent "Method and Means for Recognizing Complex Patterns" to refine outputs of various edge detectors. Hough Transform is carried out after Edge Detection to improve its results as it is tolerant of gaps in the edges, relatively unaffected by noise, occlusion in the image and robust to partial deformation in shape as can detect multiple occurrences of a shape in the same pass.It was brought to attention of the mainstream image processing community by [8].Then Duda et al. in [9] not only introduced the polar parameterization technique for more efficient line detection, but also demonstrated how a circle can be detected. Kimme et al. in [10] made circular curve detection significantly more effective by using the gradient information of pixels. It showed how the HT could be generalized to detect an arbitrary shape at a given orientation and a given scale. Ballard et al. in [11] eventually generalized the HT to detect curves of a given arbitrary shape for any orientation and any scale. Since then a lot of applications, variants and extensions of the HT have been published whose survey was given in [12]. Hough transform is further classified as Classical and Generalized. Classical Hough Transform can locate regular curves like straight lines, circles, parabolas, ellipses, etc. whereas Generalized Hough Transform can be used where a simple analytic description of

feature is not possible. The classical Hough transform was concerned with the identification of lines in the image as was introduced in [13] but later the Hough transform has been extended to identifying positions of arbitrary shapes as explored in [14] and also for finding circles or ellipses as was introduced in [15]. Hough Transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of this technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

II. DESIGN OF HYBRID EDGE DETECTION TECHNIQUE

In this section, a new hybrid edge detection technique for extracting edge and smooth areas from an image is proposed which integrates canny edge detection proposed by (Canny, 1986) and Enhanced Hough transform edge linking technique given by (Hough, 1962).

A. Edge Detection using Canny

Algorithm:

Step 1 (Noise Reduction): This step is pre processing step on image before carrying any actual filtering process. It is to remove any kind of noises due to environmental disturbances which could affect the filtering results. Gaussian filters because of their simplicity are used for this. A suitable mask is calculated and Gaussian smoothing is performed using standard convolution method. Convolution mask being smaller in size than that of actual image size is slid over the image square by square. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used is:

$$\frac{1}{115} * \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

and σ =1.4…….(12)

Step 2 (Compute Edge Strength): After performing preprocessing on given image for removing all types of noises, edge strength is calculated by taking the gradient of the image. Sobel operator is widely popular for this and so is used in this step. It calculates two-dimensional spatial gradient and estimates edge strength from absolute gradient magnitude at each point. A pair of 3x3 masks is used in which one provides gradient in the x-direction i.e. columns and other one in the y-direction i.e. rows. Sobel Convolution Masks are as:

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(13)

The magnitude, or edge strength, of the gradient is then approximated using the formula:

$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy|$$

Step 3 (Compute Direction of Edges): The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum of x is equal to zero so this condition is to be avoided always. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is:

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right) \quad (15)$$

Step 4 (Compute Angle θ'): Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x x x x x
x x x x x
x x a x x
x x x x x
x x x x x…….....…….(16)

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). This condition can be represented by a semicircle and dividing it into 5 regions as shown in Figure 5.

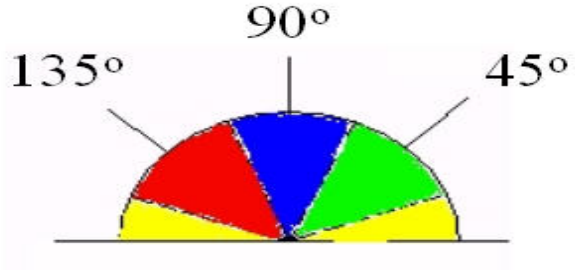

Fig. 5 Edge Direction Determination

Therefore, θ' is computed in this step. For any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees), θ' is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees), θ' is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees), θ' is set to 90 degrees. Finally, for any edge direction falling within the red range (112.5 to 157.5 degrees), θ' is set to 135 degrees.

Step 5 (Non –Maximum Suppression): After the edge directions are determined the non-maximum suppression step is applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image. The edges found after applying Sobel filter can be either very thick or very narrow depending on the intensity across the edge and how much the image was blurred. This step keeps only those pixels on an edge with the highest gradient magnitude. These maximal magnitudes should occur right at the edge boundary, and the gradient magnitude should fall off with distance from the edge.

So, three pixels in a 3 × 3 around pixel (x, y) are examined:

• If $\theta'(x, y) = 0^{\circ}$, then the pixels (x + 1, y), (x, y), and (x − 1, y) are examined.

• If $\theta'(x, y) = 90^{\circ}$, then the pixels (x, y + 1), (x, y), and (x, y − 1) are examined.

• If $\theta'(x, y) = 45^{\circ}$, then the pixels (x + 1, y + 1), (x, y), and (x − 1, y − 1) are examined.

• If $\theta'(x, y) = 135^{\circ}$, then the pixels (x + 1, y − 1), (x, y), and (x − 1, y + 1) are examined.

If pixel (x, y) has the highest gradient magnitude of the three pixels examined, it is kept as an edge. If one of the other two pixels has a higher gradient magnitude, then pixel (x, y) is not on the "center" of the edge and should not be classified as an edge pixel.

Step 6 (Hysteresis Thresholding): Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. Hysteresis is one way of solving this problem. Instead of choosing a single threshold, two thresholds TH (T-HIGH) and TL (T-LOW) are used. Pixels with a gradient magnitude (GM) < TL are discarded immediately. However, pixels with TL ≤ GM < TH is only kept if they form a continuous edge line with pixels with high gradient magnitude (i.e., above TH).So this step is carried on using following conventions:

• If pixel (x, y) has gradient magnitude less than TL, discard the edge (write out black).

• If pixel (x, y) has gradient magnitude greater than TH, keep the edge (write out white).

• If pixel (x, y) has gradient magnitude between TL and TH and any of its neighbors in a 3 × 3 region around it have gradient magnitudes greater than TH, keep the edge (write out white).

• If none of pixel (x, y)'s neighbors have high gradient magnitudes but at least one falls between TL and TH, search the 5 × 5 region to see if any of these pixels have a magnitude greater than TH. If so, keep the edge (write out white).

• Else, discard the edge (write out black).

B. Line/Edge Linking/Boundary/Circle Detection using Enhanced Hough Transform

We have used Enhanced Hough transform which combines both classical and generalized Hough transform to extract lines, edge boundaries and circles.

Algorithm for Proposed Enhanced Hough Transform

Algorithm for Line detection: Edge detection is often used as preprocessing to Hough transform. The input image must be a thresholded edge image. The magnitude results computed by the canny operator can be thresholded and used as input.

Pre-steps before applying line detection algorithm are:

1. Parameterize the line

$$y = ax + b \quad (17)$$

In the parameter spaces determined by a and b, each parameter pair (a, b) represents a line. Alternatively, we can also parameterize a line in polar coordinates as

$$x\cos\theta + y\sin\theta - \rho = 0 \quad (18)$$

2. Quantize parameters ρ and θ.

To keep the parameter space finite, it is often necessary to quantize the parameter space. By quantization, the parameter space is divided into finite grid of cells; the dimensions of each cell are determined by the quantization intervals. Each cell is indexed by a pair of quantized (θ, ρ).

3. Build an accumulator array A (θ, ρ).The accumulator array is used to accrue the contribution of each image point to a quantized (θ, ρ). The standard HT scheme for updating the accumulator array A(θ, ρ) is $A(\theta, \rho) = \#\{(x, y) \text{ belongs to } X \times Y \mid x\cos\theta + y\sin\theta + \rho = 0\}$ (19)

where X × Y are the image domain.

Algorithmic steps:

For an input gray scale or color image I of M × N.

1. Locate the Hough Transform coordinate system

2. Identify the ranges for θ and ρ. Let the range for θ be between θl and θh, and the range for ρ between ρl and ρh.

3. Choose quantization intervals δθ and δρ for θ and ρ respectively.

4. Discretize the parameter space of ρ and θ using sampling steps δρ and δθ. Let θd and ρd be one dimensional arrays containing the discretized θ and ρ.

5. Let A (T, R) be an array of integer counter; initialize all elements of A to zero, where R =ρh-ρl/δρ and and T = θh-θl/δθ.

6. Let L (T, R) be an array of a list of 2D coordinates.

7. For each image pixel (c, r), if its gradient magnitude g(c, r) > τ, where τ is a gradient magnitude threshold.

for i=1 to T

$$\rho = c * \cos(\theta d(i)) + r * \sin(\theta d(i)) \quad (20)$$

- find the index k, for the element of ρd closest to ρ
- increment A(i,k) by one.
- Add point (c,r) to L(i,k)

8. Find all local A(ip, kp) such that A(ip, kp) > t, where t is a threshold.

9. The output is a set of pairs (θd(ip), ρd(kp)) and lists of points in L(ip, kp), describing the lines detected in the image, along with points located on these lines.

Algorithm for Edge linking

1. Obtain a thresholded edge image
2. Specify subdivisions in the ρθ-plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Continuity here normally means distance between disconnected pixels and a gap in the line can be bridged if the length of the gap is less than a certain threshold.

Algorithm for Circle detection

A circle in the xy-plane is given by

$$x-a)2+(y-b)2=c2 \quad (21)$$

• So we have a 3D parameter space.
• Simple procedure:

```
set all A[a,b,c]=0
for every (x,y) where g(x,y)>T
for all a and b
        c=sqrt((x-a)^2+(y-b)^2);          (22)
A[a,b,c] = A[a,b,c]+1;
```

The outcome of Hough transform is shown below in Figure 6.

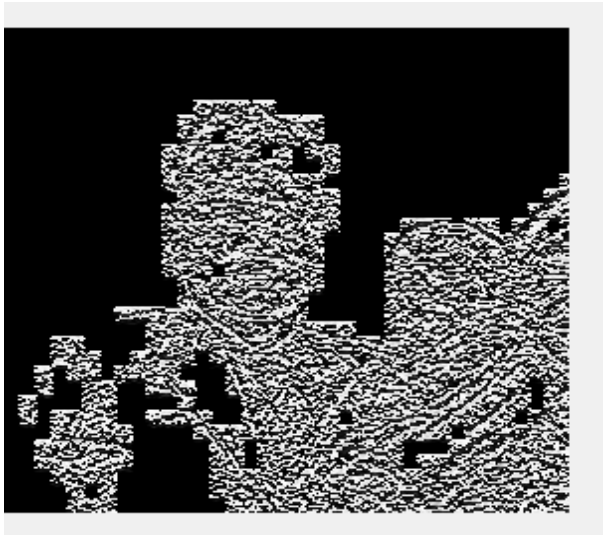

Fig. 6 Outcome of Hough Transform

The proposed feature detector would firstly apply canny on input bitmap image whose output would be further refined by applying proposed enhanced hough transform to overcome the flaws of canny as shown in Figure 7.

Advantages of new hybrid edge detector

It is using probability for finding error rate.

It is easier to locate various features like lines, edges, triangle, circles and boundaries with same accuracy and efficiency with this method.

It helped to improve signal to noise ratio which was one of the main objective of this research.

It is resistant to all kinds of noise, disturbances and provides good detection.

It is tolerant towards holes in the boundary line.

Implementation

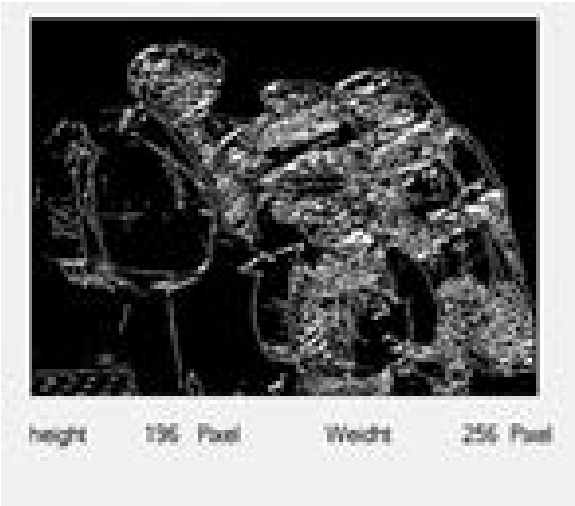

Fig. 7 Outcome of New Proposed Feature Detector

III. IMPLEMENTATION OF HYBRID EDGE DETECTION TECHNIQUE

A. Procedure for edge detection

1. The algorithm reads a 24-bit color image denoted by CI= {cp1, cp2, cp3, cpn-1} where cpi is the ith pixel in the image and n is the total number of pixels.

In the same context, every color pixel cpi can be represented as cpi={rci[rc0,….rc7], gci[gc0,…..gc7], bci[bc0,……bc7]} where i is the index of the ith pixel, rci is the ith bit of color component rc, gci is the ith bit of color component gc, and bci is the ith bit of color component bc. Here CI is a 24-bit image so each of its pixels is made up of three color components each of which is of length 8 bits.

2. The algorithm converts CI into a Gray image denoted by f(CI)=GI where GI is gray image. The purpose of this conversion is to ease the processing of subsequent steps.

3. Three parameters 1) the size of the Gaussian filter, 2) a low threshold, and 3) a high threshold are automatically chosen where results of filter are optimal.

4. The Canny edge detection algorithm is executed on GI as described in Section II using the three parameters selected in step 3. The results are a collection of lines, curves, and points denoting the edges or the boundaries of the objects in the image GI. The pixels that constitute the extracted edges are represented as EP= {ep1, ep2, ep3 …epr-1} where ej is the jth pixel that makes up the edges and r is the total number of these pixels.

Thereafter we apply Enhanced Hough transform to extract various other features like shapes lines and circles as explained below.The output image after applying Canny edge detector is having distorted edges which are not properly joined with each other so edge linking is required to fill those edge gaps. Therefore a global edge linking technique i.e. Hough

Transform has been applied (as described in procedure 1and 2 below) on the output image obtained by canny so as to get more refined edges. Hough Transform would even detect line, edges, linkings and shapes which were not traceable through Canny. The edge pixels retrieved after applying Hough Transform are represented as

HEP= {hep1, hep2, hep3 …hepq-1} where hepj is the jth pixel in the image and q is total number of edge pixels.

1. Lines/Edge Linking/boundaries Detection

Hough transform is used to refine output of canny edge detector as explained in Section 3.1.It is used to detect peaks, edge links, lines and circles in the output image retrieved from canny edge detector**.**

a. **Conversion from xy plane to ρθ plane:** This function used to transform xy plane to ρθ plane in which Hough transform works.

Syntax:

[htm, θ, ρ] = convert_xy_ ρθ (eps, δθ, δρ)

Inputs: eps is the edge pixels image retrieved after applying canny edge detector on given input cover image. δθ and. δρ represents the Hough transform bins along the θ axis ρ axis respectively

Outputs: convert_xy_ρθ () converts xy-plane to ρθ-plane. The output obtained after carrying this plane transformation is represented by a matrix named Hough transform matrix denoted by htm. This output Hough transform matrix is nρ x nθ in which nρ = 2*ceil (norm (size (eps))/ δρ) -1 and nθ = 2*ceil(90/ δθ). Here θ and ρ are element vectors specifying the angle in degree corresponding to each column and row of htm respectively.

b. **Peak Detection:** Function hough_peaks_detect () is used to detect peaks in Hough transform matrix retrieved in a. The steps followed are as:

i. Locate all those cells in Hough transform matrix with maximum value as compared with other cells and store their positions.

ii. Suppress all rest of Hough transform cells of Hough transform matrix which are in neighborhood with cells in step i to zero.

iii. Repeat steps i and ii until desired threshold is reached or target number of peaks is not reached.

Syntax:

[row_cord, col_cord, htm_p]= hough_peaks_detect (htm, max_num_of_peaks, threshold, neighborhood)

Inputs: Max_num_of_peaks specifies the maximum number of peaks to be traced from htm (Output matrix from step a). Any value for Hough transform cell in htm below given threshold value is not to be taken as peak. neighborhood is a two elements vector that defines the location of Hough transform cells in htm to be provided with suppression. These are the cells around each identified peak which are suppressed to zero.

Outputs: htm_p is output matrix successfully representing all peaks locations along with suppressed neighborhood. Each and every peak is identified by their row and column coordinates row_cord and col_cord respectively.

c. Line Detection and Linking:

a. After successful detection of all peaks in hough transform matrix, next step is to determine line segments associated with those peaks along with the start and end of those segments. So Locations of all non-zero pixels which formulate that peak are determined. The same is done using hough_pixels_detect ().

Syntax:

[row_cord, col_cord] = hough_pixels_detect (htm_p, θ, ρ, row_bins, col_bins)

Inputs: htm_p is Hough transform matrix with all peaks locations returned from step b; θ and ρ are the axis to work in; row_bins and col_bins are row and columns bins positions returned by hough_peaks_detect().

Outputs: hough_pixels_detect returns the row and column coordinates for all pixels that formulate the line segments.

b. The next step is to trace line segments formulated from the pixels determined in step a. The pixel coordinates returned by using hough_pixels_detect () are clubbed together to formulate line segment. Function hough_lines_detect () work in followed steps:

1. Rotate the pixels along vertical line.
2. Sort the pixel locations.
3. Locate gaps in different line segments and merge line segments whose gap is less than mingap specified.
4. Return only those line segments with the length greater than minimum length specified.

Syntax:

Lines_detected = hough_lines_detect (htm_p, θ, ρ, row_bins, col_bins, min_gap_in_lnsegments, minimal_length)

Inputs: htm_p is Hough transform matrix with all peaks locations returned from step a; θ and ρ are axis to work in given by function convert_xy_ ρθ (); row_bins and col_bins are rows and columns bins positions returned by hough_peaks_detect ().

Outputs: Line segments

hough_lines_detect () detects all line segments in the output matrix htm_p traced using given row_bins and col_bins. If any two retrieved line segments from same Hough transform bin are disjoint by a gap of less than specified min_gap_in_lnsegments are combined to single segment .Now, if the length of this newly formulated segment is less than minimal_length all pixels forming it are discarded.

2. Circle Detection

Circle detection is done using function hough_circle_detection () which detects multiple circles in an image using Hough transform.

Syntax:

Circles = hough_circle_detection (eps, min_rad, max_rad, min_edge_pxs_ratio, max_circle_diff);

Inputs: Here eps is input image for hough_circle_detection () function which is retrieved after application of canny edge detector on input cover image and so is gray image. min_rad is minimal radius possible and max_rad is maximum radius

possible for circle in pixels. Min_edge_pxs_ratio is the ratio of minimal number of detected edge pixels of a circle to the circle perimeter. It is generally between 0 to1 (0<min_edge_pxs_ratio>1) and its default value taken is 0.3 and so is optional argument .max_circle_diff is the maximum possible difference between two circles for them to be considered as the same one whose default value taken is 12 and so is an optional argument.; For example if fst_circle (p1, q1, r1) and sec_circle (p2, q2, r2) are the two circles detected then max_circle_diff = |p1-p2|+|q1-q2|+|r1-r2|.

Outputs: It is m-by-4 array of m circles (p, q, r, no_of_pixels) where p and q are center coordinates and r is radius and no_of _pixels give the count of pixels covered.

The step wise description for hough_circle_detection () is as follows:

a. Input validation step:

- If argument max_circle_diff is missing use its default value 12.
- If min_edge_pxs_ratio is missing use its default value 0.3.
- If the number of arguments<3, raise an error.
- Make sure all argument values are positive.

b. Create a Three Dimensional Hough array in which first two dimensions specify the coordinates of the circle center and the third argument specify the radius.

c. Edges are detected using Canny edge detector while resetting the value for threshold to balance between the performance and detection quality.

d. (ep_x-max_rad to ep_x+max_rad, ep_y-max_rad to ep_y-max_rad) are the possible locations for circle centers for an edge pixel (ep_x, ep_y) so mapping is done in same manner.

e. Formulate grid (0 to d_max_rad, 0 to d_max_rad), and then compute the distances between the center and all the grid points to form a radius. Here d_max_rad is double the maximum radius max_rad.

f. Increment the corresponding elements in the Hough array for each edge pixel determination.

g. Mark these circles highlighted on input cover image while deleting duplications.

Results of application of new hybrid filter on family and pepper images is shown in Figure 8 and Figure 9.

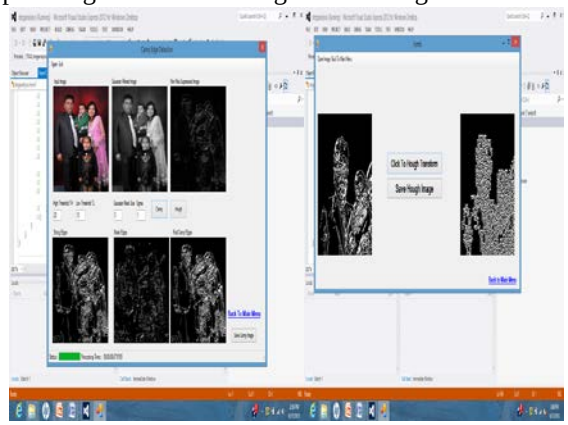

Fig. 8 Applying hybrid filter on Family.bmp

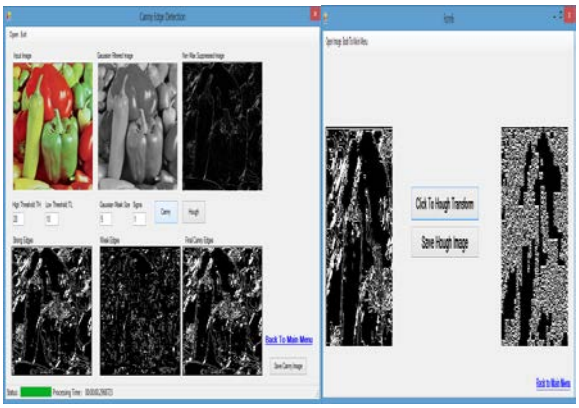

Fig. 9 Applying hybrid filter on Pepper.bmp

IV. Conclusion

A hybrid edge detection technique based on canny and hough transform is proposed in this paper. Apart from edge detection, the proposed filter is able to extract lines, boundaries and circles from given image. As per comparison analysis of various edge detectors in the present work, canny edge detector is better than all other detectors but is not that immune to noise and disturbances. Edges extracted from canny filter are refined with the application of hough transform which makes the proposed filter immune to various disturbances.

References

[1] Gonzalez, R.C., and Woods, R.E., (2003), "Digital Image Processing", Pearson Education.

[2] Albovik (2000), "Handbook of Image and Video Processing", Academic Press.

[3] Canny, J. F., (1986), "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), pp. 679-697.

[4] Canny Edge Detection ,09gr820, March 23, 2009.

[5] Heath, M., Sarkar, S., Sanocki, T., and Bowyery, K., (January, 1998), "Comparison of Edge Detectors: A Methodology and Initial Study", Computer Vision and Image Understanding, 69(1), pp. 38–54.

[6] Mamta Juneja , Parvinder Singh Sandhu, (December, 2009), " Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain", International Journal of Computer Theory and Engineering, 1(5), pp.1793-8201

[7] Hough, P.V.C., (1962), "Method and Means for Recognizing Complex Patterns", U.S. Patent 3069654.

[8] Rosenfled, A., (1969), Picture Processing by computer, Academics Press, New York.

[9] Duda, R. O. and Hart, P. E., (January, 1972), "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Communication of the ACM, 15(1), pp. 11–15.

[10] Kimme, C., Ballard, D., and Sklansky, J., (February, 1975), "Finding Circles by an Array of Accumulators", Communication of the ACM, 18(2).

[11] Ballard, D.H., (1981), "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, 13(2), pp. 111-122.

[12] Illingworth, J., Kittler, J., (October, 1988), "A Survey of the Hough Transform", Computer Vision Graphics and Image Processing, 44(1), pp. 87–116

[13] Yip, R.K.K., (August 22-26, 1994), "Line Patterns Hough Transform for Line Segment Detection", Proc. IEEE Ninth Annual International Conference on Frontiers of Computer Technology, (TENCON '94), vol.1, pp.319-323.

[14] Leavers, V.F.F., (1992), "Shape Detection in Computer Vision Using the Hough Transform", New York, Springer-Verlag.

[15] Ioannou, D., Huda, W. and Laine, F., (1999), "Circle Recognition through a 2D Hough Transform and Radius Histogramming", Image and Vision Computing, 17(1999), pp. 15-26.

Ms. Mamta Juneja is an Assistant Professor in University Institute of Engineering and Technology, Panjab University, Chandigarh, India.. She did masters in Computer Science from Punjab Technical University, India and currently pursuing Doctorate in the same. Her interest areas include Image Processing, Steganography, Information Hiding and Information Security.
Email:er_mamta@yahoo.com

Prof. Dr. Parvinder S. Sandhu is Doctorate in Computer Science and Engineering and working as Professor in Computer Science & Engineering department at Rayat & Bahra Institute of Engineering and Bio-Technology, Mohali, Punjab, INDIA. He is editorial committee member of various International Journals and conferences. He has published more than 150 research papers in various referred International journals and conferences. He chaired more than 100 renowned International Conferences and also acted as keynote speaker in different countries. His current research interests are Software Reusability, Software Maintenance, Machine Learning and Image Processing. Email: parvinder.sandhu@gmail.com